

EXPRESS MAIL NO.: EV396290662US

DATE OF DEPOSIT: January 16, 2004

This paper and fee are being deposited with the U.S. Postal Service Express Mail Post Office to Addressee service under 37 CFR §1.10 on the date indicated above and is addressed to: Mail Stop PATENT APPLICATION, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Nishi PASAREYA



Name of person mailing paper and fee

Signature of person mailing paper and fee

**INFORMATION HANDLING SYSTEM  
CAPABLE OF OPERATION IN REDUCED POWER STATES**


Inventors: Craig Chaiken  
1102 Vanderbilt Circle  
Pflugerville, Texas 78660

Muhammed K. Jaber  
40 Forest Mesa  
Round Rock, Texas 78664

Adolfo S. Montero  
3409 Gable Drive  
Austin, Texas 78759

Assignee: Dell Products L.P.  
One Dell Way  
Round Rock, Texas 78682

David L. McCombs  
HAYNES AND BOONE, L.L.P.  
901 Main Street  
Suite 3100  
Dallas, Texas 75202-3789  
Telephone: (214) 651-5533  
Facsimile: (214) 200-0853  
e-mail: [ipdocketing@haynesboone.com](mailto:ipdocketing@haynesboone.com)

|   |  |
|---|--|
| EXPRESS MAIL NO.: <u>EV396290662US</u>  | DATE OF DEPOSIT: <u>January 16, 2004</u>   |
| This paper and fee are being deposited with the U.S. Postal Service Express Mail Post Office to Addressee service under 37 CFR §1.10 on the date indicated above and is addressed to: Mail Stop PATENT APPLICATION, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450. |  |
| <u>NISHI PASAYYA</u>  |  |
| Name of person mailing paper and fee  | Signature of person mailing paper and fee  |

**INFORMATION HANDLING SYSTEM  
CAPABLE OF OPERATION IN REDUCED POWER STATES**

Background

[0001] The present disclosure relates generally to information handling systems (IHS's), and more particularly to information handling systems which feature reduced power consumption.

[0002] As the value and use of information continues to increase, individuals and businesses seek additional ways to process and store information. One option available to users is information handling systems. An information handling system generally processes, compiles, stores, and/or communicates information or data for business, personal, or other purposes thereby allowing users to take advantage of the value of the information. Because technology and information handling needs and requirements vary between different users or applications, information handling systems may also vary regarding what information is handled, how the information is handled, how much information is processed, stored, or communicated, and how quickly and efficiently the information may be processed, stored, or communicated. The variations in information handling systems allow for information handling systems to be general or configured for a specific user or specific use such as financial transaction processing, airline reservations, enterprise data storage, or global communications. In addition, information handling systems may include a

variety of hardware and software components that may be configured to process, store, and communicate information and may include one or more computer systems, data storage systems, and networking systems.

**[0003]** As power conservation is an ever present issue, many modern IHS's are designed with power savings features. A common method used in IHS's to achieve lower power usage is to configure IHS's and components with multiple power states. For example, according to the current Advanced Configuration and Power Interface (ACPI) standard, a typical IHS may enter into power savings states ranging from S0 (full on) to S4 (hibernate). More relevant to this disclosure, Intel Architecture based processors may operate in various power savings states known as "C states", ranging from C0 (full power – least power saving state) to C4 (highest power saving state). Because "C States" higher than C2 present significantly greater power savings, it would be desirable for processors to operate in C3 and higher states whenever possible.

**[0004]** IHS's which feature bus mastering components present significant hurdles to achieving C3 and higher power savings states because any bus mastering activity prevents processors from entering C3 and higher states. One example of a type of component which utilizes bus mastering is Universal Serial Bus (USB). USB component's prevalence in modern IHS's presents an acute challenge in designing more power efficient IHS's.

**[0005]** Some existing USB devices support power savings states through what is known as "selective-suspend." "Selective-suspend" allows USB hubs to be suspended if no devices are connected to the hub. In addition, if only suspended USB devices are connected to a USB controller, the controller may also be suspended. Therefore, IHS's equipped with only "selective-suspend" compliant USB devices may operate with their processors in C3 and higher states. However, many

USB devices do not support “selective-suspend.” This is especially true for USB storage devices.

**[0006]** Therefore, what is needed is a technique for enabling an IHS equipped with a bus mastering device, such as a non selective-suspend compliant USB device, to operate with its processor in enhanced power savings states such as C3 and higher states to reduce power consumption.

### Summary

**[0007]** Accordingly, in one embodiment, a method for allowing a processor to enter low power states in an IHS is disclosed which includes detecting an access request for a bus mastering device. The method also includes in response to failing to detect an access request for the bus mastering device within a predetermined period of time, suspending a bus mastering device controller associated with the bus mastering device, wherein the now suspended bus mastering controller no longer prevents the processor from entering low power states.

**[0008]** In another embodiment, an information handling system (IHS) is disclosed which includes a processor capable of entering low power states. The IHS includes a memory coupled to the processor and a non-volatile storage coupled to the processor. The IHS also includes a bus mastering device and a bus mastering device controller coupled to the bus mastering device and the processor for transferring information between the bus mastering device and the processor. Executable code is stored in the non-volatile storage for detecting an access request for the bus mastering device and causing the bus mastering device controller to be suspended in response to failing to detect an access request for the bus mastering device within a predetermined period of time, wherein the now suspended controller no longer prevents the processor from entering low power states.

**[0009]** A principal advantage of the embodiment disclosed herein is that the IHS can operate with its processor in low power states, resulting in reduction in power consumption.

### Brief Description of the Drawings

**[0010]** Fig. 1 is a block diagram of the disclosed information handling system.

**[0011]** Fig. 2A is a flow chart showing the filter driver initialization process in the information handling system of Fig. 1.

**[0012]** Fig. 2B is a flow chart showing the process flow of suspending the USB controller in the information handling system of Fig. 1.

**[0013]** Fig. 2C is a flow chart showing the process flow of resuming operation of the USB controller in the information handling system of Fig. 1.

### Detailed Description

**[0014]** For purposes of this disclosure, an information handling system may include any instrumentality or aggregate of instrumentalities operable to compute, classify, process, transmit, receive, retrieve, originate, switch, store, display, manifest, detect, record, reproduce, handle, or utilize any form of information, intelligence, or data for business, scientific, control, or other purposes. For example, an information handling system may be a personal computer, a network storage device, or any other suitable device and may vary in size, shape, performance, functionality, and price. The information handling system may include random access memory (RAM), one or more processing resources such as a central processing unit (CPU) or hardware or software control logic, ROM, and/or other types of nonvolatile memory. Additional components of the information handling

system may include one or more disk drives, one or more network ports for communicating with external devices as well as various input and output (I/O) devices, such as a keyboard, a mouse, and a video display. The information handling system may also include one or more buses operable to transmit communications between the various hardware components.

**[0015]** In one embodiment, information handling system 100, Fig. 1, includes a processor 105 such as an Intel Pentium series processor. Processor 105 is capable of operating in one of the above mentioned "C states" to conserve power. An Intel Hub Architecture (IHA) chipset 110 provides IHS system 100 with graphics/memory controller hub functions and I/O functions. More specifically, IHA chipset 110 acts as a host controller which communicates with a graphics controller 115 coupled thereto. A display 120 is coupled to graphics controller 115. Chipset 110 further acts as a controller for main memory 130 which is coupled thereto. Chipset 110 also acts as an I/O controller hub (ICH) which performs I/O functions. A super input/output (I/O) controller 140 is coupled to chipset 110 to provide communications between chipset 110 and input devices 145 such as a mouse, keyboard, and tablet, for example. A USB controller 152 is coupled to chipset 110 so that devices such as a USB device 154 can be connected to chipset 110 and processor 105. USB device 154 is coupled to USB controller 152 via USB 150. USB devices that may be coupled to USB controller 152 include floppy disk drives, CD-ROM drives, DVD-ROM drives and other devices which support the USB standard. System basic input-output system (BIOS) 160 is coupled to chipset 110 as shown. BIOS 160 is stored in CMOS or FLASH memory so that it is nonvolatile.

**[0016]** A local area network (LAN) controller 170, alternatively called a network interface controller (NIC), is coupled to chipset 110 to facilitate connection of system 100 to other information handling systems. Media driver controller 180 is coupled to chipset 110 so that devices such as media drives 185 can be connected to chipset

110 and processor 105. Devices that can be coupled to media controller 180 include CD-ROM drives, DVD drives, hard disk drives and other fixed or removable media drives. An expansion bus 190, such as a peripheral component interconnect (PCI) bus, PCI Express bus, serial advanced technology attachment (SATA) bus or other bus is coupled to chipset 110 as shown. Expansion bus 190 includes one or more expansion slots (not shown) for receiving expansion cards which provide IHS 100 with additional functionality.

**[0017]** USB 150, USB controller 152, and USB device 154 feature bus mastering. USB device 154 is an example of a bus mastering device and USB controller 152 is an example of a bus mastering device controller. As discussed earlier, in conventional systems, any bus mastering activity prevents processor 105 from being placed into states C3 and higher. Also as mentioned above, although “selective suspend” allows some USB equipped IHS’s to place their processors in C3 and higher states, all USB devices in an IHS must support “selective suspend” in order for this feature to operate. Here, USB device 154 is not a “selective suspend” compliant device. To remedy the power issue related to devices featuring bus mastering, and particularly in this embodiment with USB devices, IHS 100 includes additional features as described herein.

**[0018]** While IHS 100 is actively operational, an application 136 is loaded in main memory 130. Application 136 may be a word processing application, graphics editor, or any other available software application. Also loaded in main memory 130 is an operating system (OS) 132, such as one of Microsoft Windows family of operating systems. (Microsoft and Windows are trademarks of Microsoft Corp.). In this embodiment, OS 132 includes a USB stack 133 and a filter driver 134 which is stored in the form of executable code. Main memory further includes a USB state flag 138. Although USB state flag 138 is shown to be loaded in main memory 130

here, in other embodiments, USB state flag may be stored in other types of volatile and non-volatile storage devices such as media drives 185 or BIOS 160.

**[0019]** Operation of IHS 100 may be seen by examining Figs. 2A, 2B, and 2C. Fig. 2A is a flow chart depicting process flow that occurs when filter driver 134 initializes. In block 200, filter driver 134 is loaded into main memory 130. Subsequently in block 205, filter driver 134 starts a timer (not shown) for a period of time. The period of time represents an amount of time that USB controller 152 may remain inactive before a process to suspend USB controller 152 initiates. For example, the timer may be set for 10 seconds, and if USB controller 152 remains inactive for 10 seconds, a process will begin to suspend USB controller 152. In block 210, filter driver 134 clears USB state flag 138. USB state flag 138 may either be cleared which is its default state, or in the alternative, set. Functions of USB state flag 138 as well as the timer are discussed in more detail later in this disclosure.

**[0020]** Fig. 2B is a flow chart illustrating the steps IHS 100 takes to suspend USB controller 152 in the event the aforementioned timer expires. As discussed above, the timer expires if USB controller 152 remains inactive for the period of time for which the timer is set. IHS 100 may be configured to detect USB controller 152's inactivity in a number of different ways. In this particular embodiment, the executable code of filter driver 134 is configured to monitor input/output request packets (IRP's) generated for USB controller 152. IRP's are generated by OS 132 in response to requests made by OS 132 itself or by application 136, to transmit or receive information to or from devices coupled to USB controller 152, such as USB device 154. Using a word processing application and a floppy disk drive as examples of application 136 and USB device 154, respectively, when an IHS user issues a command to save a document to a disk inserted into the floppy disk drive, the word processing application causes OS 132 to generate an IRP in performing



the task. IRP's are then transmitted to USB stack 133 for further processing. While in transmission, IRP's are detected by filter driver 134. As discussed in more detail later, detection of IRP's by filter driver 134 indicates that USB controller 152 is currently active, and the timer is restarted. Of course, restarting the timer prevents it from expiring.

**[0021]** If filter driver 134 fails to detect an IRP during the period of time as set by the timer, the timer expires. In the event that the timer expires as shown in block 220, filter driver 134 determines whether USB controller 152 is currently on (i.e., not suspended) as shown in decision block 225. If filter driver 134 determines that USB controller 152 is currently not in suspend mode, filter driver 134 causes USB controller 152 to be placed into suspend mode as shown in block 230. Various implementations are possible to cause USB controller 152 to be placed into suspend mode. In one particular embodiment, filter driver 134 may cause a system management interrupt (SMI) to be generated, which in turn places USB controller 152 into suspend mode. In addition, filter driver 134 may similarly place USB device 154 into suspend mode prior to placing controller 152 into suspend mode.

**[0022]** As shown in block 235, filter driver 134 also sets USB state flag 138. As mentioned earlier, USB state flag 138 may be set or in the alternative, cleared. When USB flag 138 is set, it functions as an indication that USB controller 152 is suspended and that when filter driver 134 needs to resume operation of controller 152 (for example, because it now detects an IRP destined for USB controller 152), it is permitted to do so. Essentially, setting USB flag 138 is an indication that filter driver 134 and not another component of IHS 100 caused USB controller 152 to be placed into suspend mode. In the present disclosure, USB controller 152 is placed into suspend mode by filter driver 134. However, in the course of IHS 100's operation, controller 152 may be placed into suspend mode by various other components of IHS 100, such as other components of OS 132. It is desirable to

prevent filter driver 134 from resuming operation of USB controller 152 that has been placed into suspend mode by another component of IHS 100. Accordingly, when the time arrives for resuming operation of USB controller 152, USB state flag 138 aids filter driver 138 in distinguishing between situations where it may perform the operation (when it is set), and situations where it must defer (when it is cleared) to the component which originally placed controller 152 into suspend mode. The process of resuming operation of controller 152 is discussed in more detail later in this disclosure.

**[0023]** At the completion of above procedures, USB controller 152, now suspended, no longer prevents processor 105 from entering into C3 and higher states, as shown in block 240. Placing USB controller 152 into suspend mode stops bus mastering activities normally engaged by controller 152. Note that if the test conducted at decision block 225 determines that USB controller 152 is already in suspend mode, then the above steps are skipped as shown in Fig. 2B and the flow directly continues to block 240.

**[0024]** Fig. 2C is a flow chart describing the process flow of resuming operation of USB controller 152. The process is initiated when OS 132 generates an IRP for USB controller 152 during the period of time in which the timer has not expired. As shown in block 250, the IRP is detected, and in this particular embodiment, filter driver 134 performs the detection. Detection of the IRP by filter driver 134 indicates a new activity, and filter driver 134 restarts the timer for the period time as illustrated in block 255 to begin measuring time of inactivity anew.

**[0025]** In decision block 260, a test is conducted to determine whether USB state flag 138 is set. As noted earlier, when USB state flag is set, it is an indication that USB controller 152 is currently in suspend mode. It is also an indication that filter driver 134 and not another component of IHS 100 caused USB controller 152 to be

placed into suspend mode. Consequently, if it is determined that USB state flag 138 is in fact set, filter driver 134 takes the necessary steps to resume operation of USB controller 152 as shown in block 270. In one embodiment, resuming operation of controller 152 is performed by filter driver 134 by generating a SMI, which in turn actually resumes operation of controller 152. Additionally, resuming operation of controller 152 may include resuming operation of USB device 154 coupled to controller 152. If an alternative outcome is reached in decision block 260 and USB state flag 138 is not set, then it is an indication that controller 152 was placed into suspend mode by another component of IHS 100 or that it was not placed into suspend mode at all. Accordingly, the flow continues directly to block 275 where IHS 100 proceeds with normal input/output (I/O) operation as shown, relying on another component as necessary to resume operation of USB controller 152 if controller 152 was placed in suspend mode by the other component.

**[0026]** Once filter driver 134 resumes operation of USB controller 152 in block 265, filter driver 134 clears USB state flag 138 as shown in block 270 to indicate that controller 152 is currently not in suspend mode. Finally, IHS 100 proceeds with normal I/O operation as shown in block 275.

**[0027]** Note that in the above discussion, a number of functions related to detecting IRP's, placing USB controller 152 into suspend mode, resuming operation of controller 152, and other aspects of present disclosure are incorporated into filter driver 134. However, in another embodiment, functions described therein may be performed by any one or more software components and/or hardware components so configured.

**[0028]** Although illustrative embodiments have been shown and described, a wide range of modification, change and substitution is contemplated in the foregoing disclosure and in some instances, some features of the embodiments may be

**PATENT**

Docket No.: 16356.841(DC-05832)

Customer No.: 000027683

employed without a corresponding use of other features. Accordingly, it is appropriate that the appended claims be construed broadly and in a manner consistent with the scope of the embodiments disclosed herein.